## IBM Employee Attrition using Hive and R

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfilment of the requirements to award the degree of

**Bachelor of Science**

In

**Computer Science**

**School of Engineering and Sciences**

Submitted by

**M.Harshini -AP21110010627**

**N.Ravindranath - AP21110010621**

**M.Pradeepika - AP21110010608**

**T.Chaitanya - AP21110010649**

****

Under the Guidance of

**Dr. Saleti Sumalatha**

**Asst. Professor, Dept. of. CSE**

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[November, 2024]

### Certificate

Date: 18-Nov-24

This is to certify that the work presented in this project entitled "IBM Employee Attrition Analysis Using Hive and R" has been carried out by **Dr. Saleti Sumalatha** under my supervision. The work is genuine, original, and suitable for submission to SRM University – AP for the award of Bachelor of Technology in the **School of Engineering and Sciences.**

**Supervisor**

**Dr. Saleti Sumalatha**

**Asst. Professor, Dept. of. CSE**

### Acknowledgments

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I would like to show my greatest appreciation to Dr. Saleti Sumalatha. I can’t say thank you enough for her tremendous support and help. I feel motivated and encouraged every time. Without her encouragement and guidance, this project would not have materialized.

The guidance and support received from my friends who helped me to complete this project was vital for the success of the project. I am grateful for their constant support and help

### 

**Table of Contents**

[Abstarct 5](#_Toc182849408)

[Introduction 6](#_Toc182849409)

[Problem Survey 7](#_Toc182849410)

[Dataset Description 8](#_Toc182849411)

[Implementation 11](#_Toc182849412)

### Abstarct

This project aims to analyze IBM’s HR dataset to identify factors influencing employee attrition. By leveraging Hive for data processing within a Hadoop ecosystem and R for visualization, the project seeks to uncover patterns that affect employee retention. The dataset includes various employee attributes such as demographics, job satisfaction, performance, and tenure. Understanding these factors can help predict attrition trends and inform retention strategies within an organization.

The data is first uploaded to the Hortonworks Sandbox and stored in HDFS, making it accessible for processing. Using Hive, data transformations and aggregations are performed to summarize key attributes and their relationship to attrition. Hive’s SQL-like syntax allows efficient querying and manipulation of large datasets, facilitating data analysis at scale. The results are then visualized using R to provide actionable insights in an intuitive format.

The project focuses on exploring patterns such as the impact of job role, salary, work-life balance, and job satisfaction on employee turnover. Predictive models are also developed to forecast attrition likelihood based on historical data. These insights are crucial for HR professionals in designing strategies to retain top talent and reduce turnover. By combining big data processing with statistical analysis, the project offers valuable contributions to HR decision-making processes.

Ultimately, this project demonstrates the power of combining Hadoop's data processing capabilities with data science tools to generate meaningful insights. It showcases how data-driven approaches can help organizations improve employee satisfaction and retention. The integration of Hive and R provides a scalable and efficient solution for analyzing large datasets in the context of human resources.

### Introduction

Employee attrition is a critical issue faced by organizations across industries, as high turnover can lead to increased recruitment costs, loss of expertise, and a negative impact on overall productivity. Understanding the underlying factors that influence employee attrition is essential for organizations aiming to retain their top talent and reduce operational disruptions. This project uses the IBM Employee Attrition dataset to identify and analyze key factors influencing employee turnover. By leveraging Apache Hive for data processing and R for visualization, the project aims to uncover patterns that predict attrition trends and help organizations make data-driven decisions regarding retention strategies.

The dataset contains various attributes related to employees, including demographic information, job satisfaction, performance metrics, tenure, and other factors that may contribute to attrition. To process and analyze this data at scale, the project utilizes Hive within a Hadoop ecosystem, which provides the computational power needed to handle large datasets efficiently. R is used to create interactive visualizations that provide intuitive insights into the relationships between different factors and employee retention. By combining the power of Hive for data manipulation with R for statistical analysis, this project offers a comprehensive approach to studying employee turnover.

#### 1.1 Motivation

The motivation for this project stems from the growing importance of data-driven decision-making in Human Resources (HR) practices. By predicting employee attrition, organizations can proactively address issues like job dissatisfaction, work-life imbalance, and compensation concerns before they lead to turnover. This can result in better employee engagement, improved retention strategies, and ultimately, cost savings. Furthermore, the ability to make informed decisions based on data is increasingly seen as a competitive advantage in today’s business environment, making this project highly relevant for HR professionals and business leaders.

#### 1.2 Required Environments

Colab Environment:

Python Data Visualization and Preprocessing ➢ Oracle DM Virtual Box:

To run Hortonworks Dockers Sandbox: ✓ Runs Ambari Admin, SSH Client

### Problem Survey

Employee attrition is a significant challenge for organizations, leading to high recruitment costs, decreased morale, and lost productivity. Identifying the underlying causes of attrition and predicting when employees are likely to leave is crucial for improving retention strategies. However, many organizations still rely on traditional methods, such as surveys and interviews, which may not capture the full scope of factors influencing employee turnover.

In recent years, companies have begun leveraging data analytics to better understand the complex relationship between various employee attributes and their likelihood of leaving. The IBM Employee Attrition dataset offers a rich collection of demographic, job-related, and performance-related variables, which can provide valuable insights into patterns of employee turnover. Despite this opportunity, processing and analyzing such large datasets can be challenging without the right tools and techniques, especially when dealing with a high volume of data.

Existing solutions for analyzing employee attrition often focus on basic statistical methods or simple linear models, which may not capture the full complexity of the problem. More advanced approaches, such as machine learning algorithms and big data analytics, are needed to identify hidden patterns and trends within the data. While platforms like Hadoop and Hive are increasingly used for big data processing, they require a certain level of expertise to set up and operate effectively.

This project aims to bridge this gap by using the Hortonworks Sandbox with Hive for data processing and R for advanced data analysis and visualization. By processing the IBM Employee Attrition dataset within this environment, the project seeks to identify key factors influencing attrition and create predictive models that can help organizations better understand and mitigate turnover. Through this approach, we aim to demonstrate how big data and analytics can transform HR practices and drive more effective retention strategies.

### Dataset Description

#### 2.1 About Dataset

The dataset is obtained from Kaggle. This dataset contains 23421 entries with 37 columns.

**Columns:**

* Age
* Attrition
* BusinessTravel
* DailyRate
* Department
* DistanceFromHome
* Education
* EducationField
* EmployeeCount
* EmployeeNumber
* Application ID
* EnvironmentSatisfaction
* Gender
* HourlyRate
* JobInvolvement
* JobLevel
* JobRole
* JobSatisfaction
* MaritalStatus
* MonthlyIncome
* MonthlyRate
* NumCompaniesWorked
* Over18
* OverTime
* PercentSalaryHike
* PerformanceRating
* RelationshipSatisfaction
* StandardHours
* StockOptionLevel
* TotalWorkingYears
* TrainingTimesLastYear
* WorkLifeBalance
* YearsAtCompany
* YearsInCurrentRole
* YearsSinceLastPromotion
* YearsWithCurrManager
* Employee Source

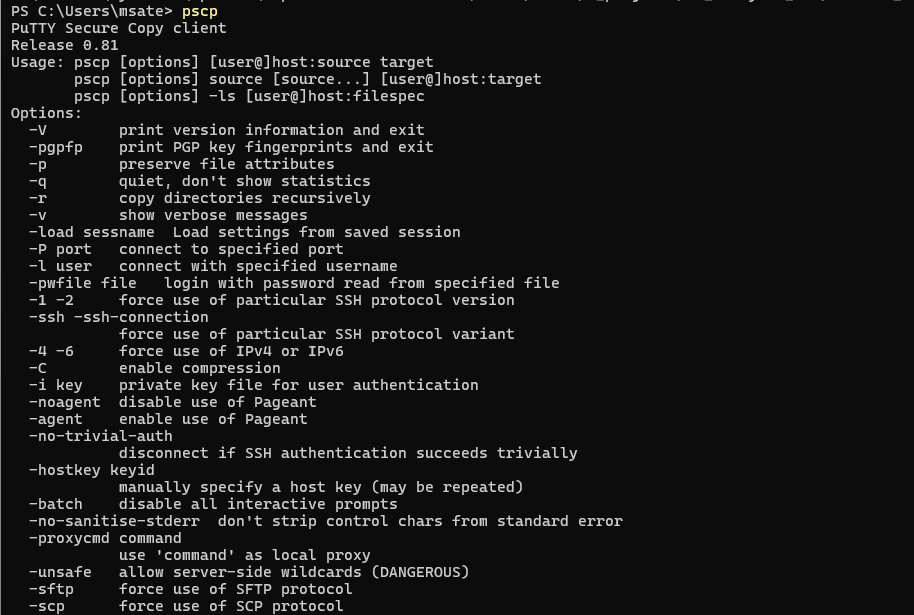
#### 2.2 Description of Dataset Attributes

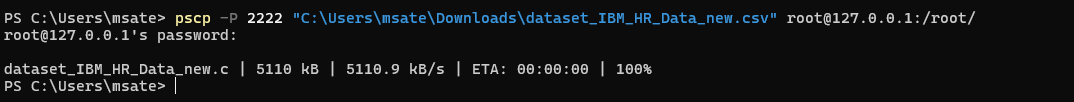
1. **Age:** Age of the employee in years.
2. **Attrition:** Indicates whether the employee has left the company (Yes/No).
3. **BusinessTravel:** Frequency of business travel (e.g., Rarely, Frequently, or Non-Travel).
4. **DailyRate:** Daily salary rate of the employee.
5. **Department:** The department the employee works in (e.g., HR, R&D, Sales).
6. **DistanceFromHome:** Distance between the employee's home and workplace
7. **Education:** Employee's highest educational level (e.g., 1=Below College, 2=College, 3=Bachelor's, etc.).
8. **EducationField:** The field of study or specialization of the employee (e.g., Life Sciences, Engineering).
9. **EmployeeCount:** Total count of employees in the dataset (often static at 1 for each row).
10. **EmployeeNumber:** Unique identifier for each employee.
11. **Application ID:** Identifier for the employee's application (if applicable)**.**
12. **EnvironmentSatisfaction:** Satisfaction level with the work environment (e.g., 1=Low, 2=Medium, etc.).
13. **Gender:** Gender of the employee (e.g., Male, Female).
14. **HourlyRate:** Hourly wage of the employee.
15. **JobInvolvement:** Measure of the employee's involvement in their job (e.g., 1=Low, 4=High).
16. **JobLevel:** The level of the employee's role within the organization hierarchy.
17. **JobRole:** Job designation or title (e.g., Manager, Research Scientist, Sales Executive).
18. **JobSatisfaction**: Employee's satisfaction with their job (e.g., 1=Low, 4=High).
19. **MaritalStatus:** Marital status of the employee (e.g., Single, Married, Divorced).
20. **MonthlyIncome:** Monthly salary of the employee.
21. **MonthlyRate:** Total monthly payment rate for the employee.
22. **NumCompaniesWorked:** No.of companies the employee has worked at previously.
23. **Over18:** Indicates whether the employee is over 18 years old (Y/N).
24. **OverTime:** Indicates whether the employee works overtime (Yes/No).
25. **PercentSalaryHike:** Percentage increase in the employee’s salary.
26. **PerformanceRating:** Employee's performance rating (e.g., 1=Low, 4=Outstanding).
27. **RelationshipSatisfaction:** Employee's satisfaction with workplace relationships (e.g., 1=Low, 4=High).
28. **StandardHours:** Standard working hours per week (e.g., typically 40 hours).
29. **StockOptionLevel:** Level of stock options granted to the employee.
30. **TotalWorkingYears:** Total years of work experience of the employee.
31. **TrainingTimesLastYear:** Number of training sessions attended in the last year.
32. **WorkLifeBalance:** Employee's perception of work-life balance (e.g., 1=Bad, 4=Excellent).
33. **YearsAtCompany:** Total years the employee has worked at the company.
34. **YearsInCurrentRole:** Total years the employee has held their current role.
35. **YearsSinceLastPromotion:** Number of years since the employee's last promotion.
36. **YearsWithCurrManager:** Number of years the employee has worked with their current manager.
37. **Employee Source:** Source of employee recruitment (e.g., Job Portal, Campus Hire)

### Implementation

#### Uploading to HDFS

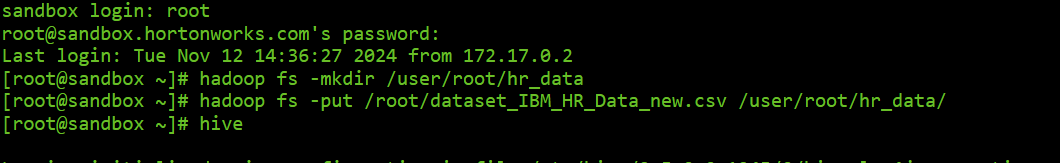
1. Create Directory in HDFS: ->(hadoop fs -mkdir /user/root/hr\_data)



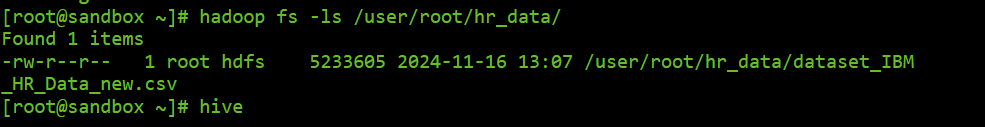


1. Move the File to HDFS:

->(hadoop fs -put /root/dataset\_IBM\_HR\_Data\_new.csv /user/root/hr\_data/)



1. Start Hive Launch Hive to begin working with your data



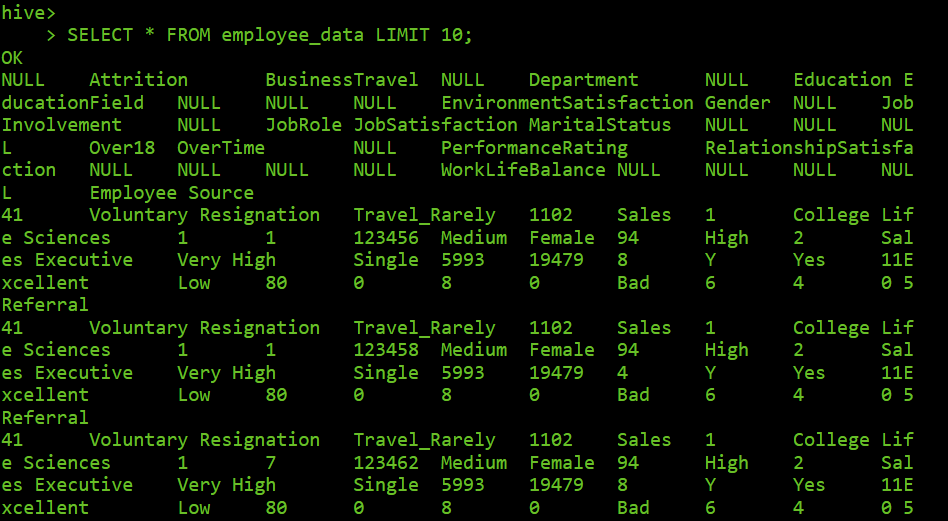
1. To check if files created and stored properly or not

->(hadoop fs -ls /user/root/hr\_data/)

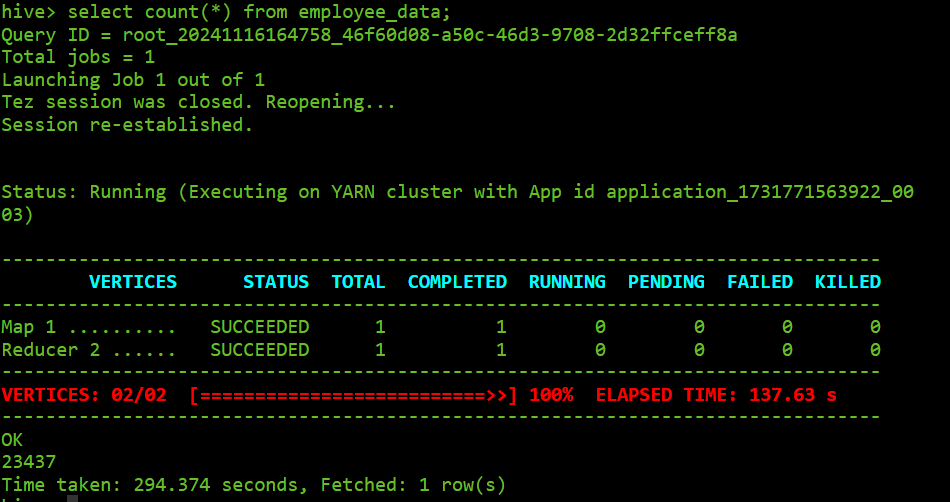
#### Hive Preprocessing :

Hive commands and their outputs:

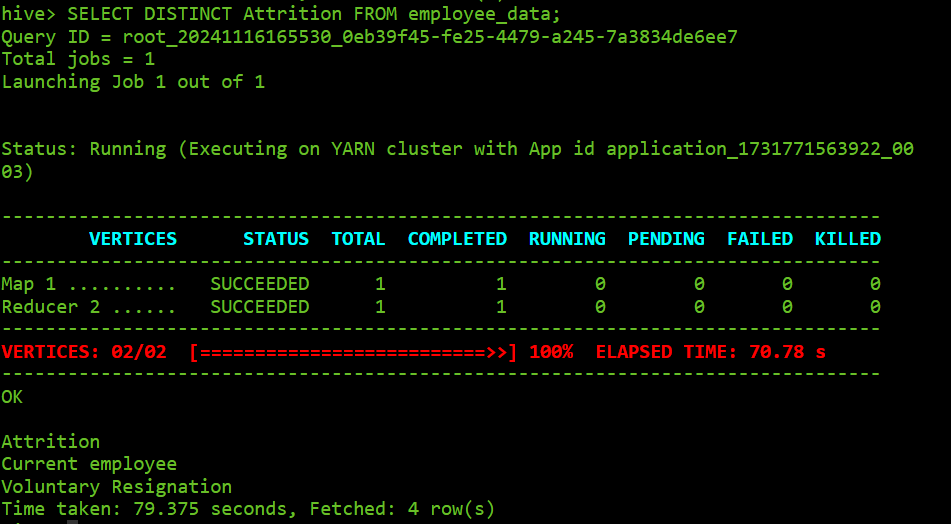
1. Check the first few rows of the dataset:



1. Total number of records



1. Distinct values in the 'Attrition' column



4)null values checking in all coulmns:

SELECT

COUNT(CASE WHEN Age IS NULL THEN 1 END) AS Age\_NULLs,

COUNT(CASE WHEN Attrition IS NULL THEN 1 END) AS Attrition\_NULLs,

COUNT(CASE WHEN BusinessTravel IS NULL THEN 1 END) AS BusinessTravel\_NULLs,

COUNT(CASE WHEN DailyRate IS NULL THEN 1 END) AS DailyRate\_NULLs,

COUNT(CASE WHEN Department IS NULL THEN 1 END) AS Department\_NULLs,

COUNT(CASE WHEN DistanceFromHome IS NULL THEN 1 END) AS DistanceFromHome\_NULLs,

COUNT(CASE WHEN Education IS NULL THEN 1 END) AS Education\_NULLs,

COUNT(CASE WHEN EducationField IS NULL THEN 1 END) AS EducationField\_NULLs,

COUNT(CASE WHEN EmployeeCount IS NULL THEN 1 END) AS EmployeeCount\_NULLs,

COUNT(CASE WHEN EmployeeNumber IS NULL THEN 1 END) AS EmployeeNumber\_NULLs,

COUNT(CASE WHEN ApplicationID IS NULL THEN 1 END) AS ApplicationID\_NULLs,

COUNT(CASE WHEN EnvironmentSatisfaction IS NULL THEN 1 END) AS EnvironmentSatisfaction\_NULLs,

COUNT(CASE WHEN Gender IS NULL THEN 1 END) AS Gender\_NULLs,

COUNT(CASE WHEN HourlyRate IS NULL THEN 1 END) AS HourlyRate\_NULLs,

COUNT(CASE WHEN JobInvolvement IS NULL THEN 1 END) AS JobInvolvement\_NULLs,

COUNT(CASE WHEN JobLevel IS NULL THEN 1 END) AS JobLevel\_NULLs,

COUNT(CASE WHEN JobRole IS NULL THEN 1 END) AS JobRole\_NULLs,

COUNT(CASE WHEN JobSatisfaction IS NULL THEN 1 END) AS JobSatisfaction\_NULLs,

COUNT(CASE WHEN MaritalStatus IS NULL THEN 1 END) AS MaritalStatus\_NULLs,

COUNT(CASE WHEN MonthlyIncome IS NULL THEN 1 END) AS MonthlyIncome\_NULLs,

COUNT(CASE WHEN MonthlyRate IS NULL THEN 1 END) AS MonthlyRate\_NULLs,

COUNT(CASE WHEN NumCompaniesWorked IS NULL THEN 1 END) AS NumCompaniesWorked\_NULLs,

COUNT(CASE WHEN Over18 IS NULL THEN 1 END) AS Over18\_NULLs,

COUNT(CASE WHEN OverTime IS NULL THEN 1 END) AS OverTime\_NULLs,

COUNT(CASE WHEN PercentSalaryHike IS NULL THEN 1 END) AS PercentSalaryHike\_NULLs,

COUNT(CASE WHEN PerformanceRating IS NULL THEN 1 END) AS PerformanceRating\_NULLs,

COUNT(CASE WHEN RelationshipSatisfaction IS NULL THEN 1 END) AS RelationshipSatisfaction\_NULLs,

COUNT(CASE WHEN StandardHours IS NULL THEN 1 END) AS StandardHours\_NULLs,

COUNT(CASE WHEN StockOptionLevel IS NULL THEN 1 END) AS StockOptionLevel\_NULLs,

COUNT(CASE WHEN TotalWorkingYears IS NULL THEN 1 END) AS TotalWorkingYears\_NULLs,

COUNT(CASE WHEN TrainingTimesLastYear IS NULL THEN 1 END) AS TrainingTimesLastYear\_NULLs,

COUNT(CASE WHEN WorkLifeBalance IS NULL THEN 1 END) AS WorkLifeBalance\_NULLs,

COUNT(CASE WHEN YearsAtCompany IS NULL THEN 1 END) AS YearsAtCompany\_NULLs,

COUNT(CASE WHEN YearsInCurrentRole IS NULL THEN 1 END) AS YearsInCurrentRole\_NULLs,

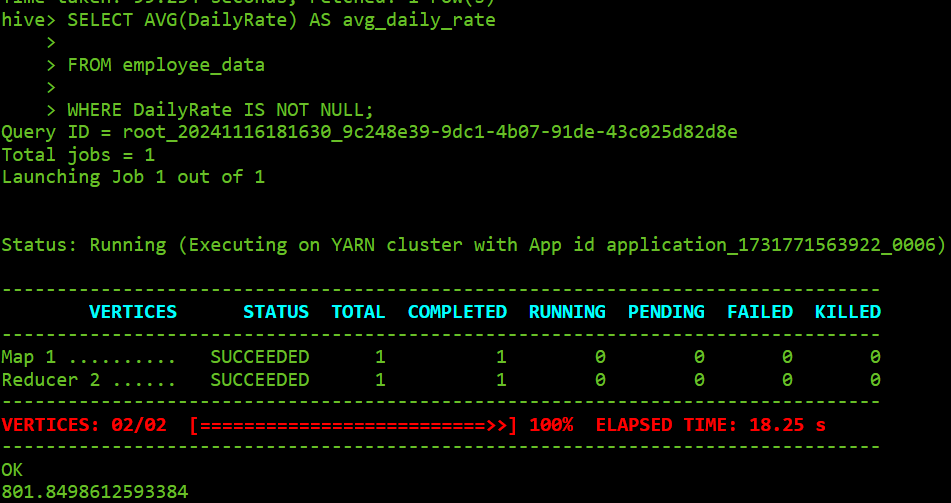
COUNT(CASE WHEN YearsSinceLastPromotion IS NULL THEN 1 END) AS YearsSinceLastPromotion\_NULLs,

COUNT(CASE WHEN YearsWithCurrManager IS NULL THEN 1 END) AS YearsWithCurrManager\_NULLs,

COUNT(CASE WHEN EmployeeSource IS NULL THEN 1 END) AS EmployeeSource\_NULLs

FROM employee\_data;

  
  
and then filling their columns with their mean ,mode and some with unknown:

->mean:  
getting mean:  


Filling with this mean:  
(here we create temp table with mean values filled ,which is used furthur)



This new table is renamed as employee table with old table deleted,  
  
so similar process is repeated for other columns,filing with mean,mode and unknown

#### R programming :

#### Load Necessary Libraries

This step loads the required R libraries for data manipulation, visualization, and machine learning.

library(dplyr)

library(caret)

library(randomForest)

library(ggplot2)

library(e1071)

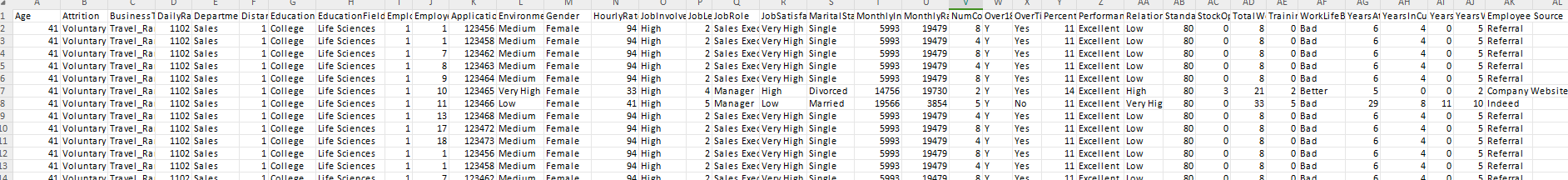
library(pROC)

library(ggcorrplot)

library(gridExtra)

**Load and Prepare Data**

data <- read.csv("C:/Users/msate/Downloads/R\_dataset.csv", stringsAsFactors = FALSE)



**Drop Unnecessary columns**

data <- data %>% select(-EmployeeNumber, -Application.ID, -StandardHours, -EmployeeCount)

**Remove Null Values**

data <- na.omit(data)

**Transform the Data into useful form**

data$Attrition <- factor(data$Attrition)

categorical\_cols <- c('BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRole', 'MaritalStatus', 'Over18',

'OverTime', 'Employee.Source', 'Education')

data[categorical\_cols] <- lapply(data[categorical\_cols], factor)

data[categorical\_cols] <- lapply(data[categorical\_cols], as.numeric)

**Visualization**

**Distribution of attrition**

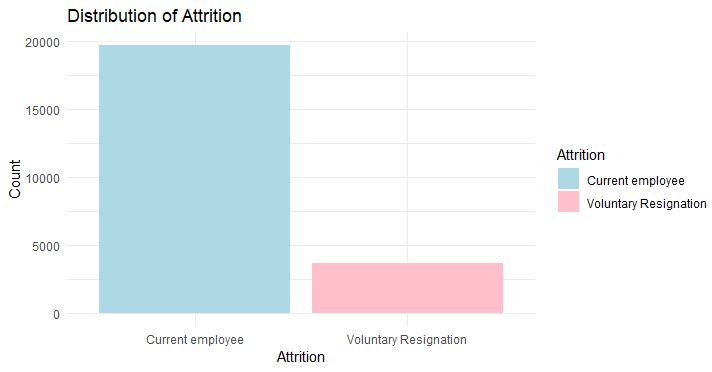
ggplot(data, aes(x = Attrition, fill = Attrition)) +

geom\_bar() +

labs(title = "Distribution of Attrition", x = "Attrition", y = "Count") +

scale\_fill\_manual(values = c("lightblue", "pink")) +

theme\_minimal()



**Attrition by Gender**

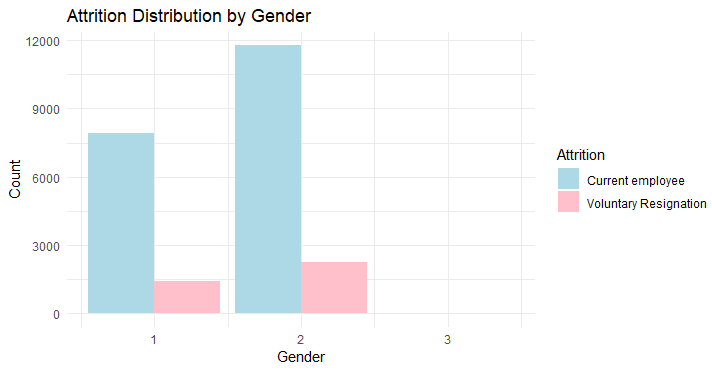
ggplot(data, aes(x = Gender, fill = Attrition)) +

geom\_bar(position = "dodge") +

labs(title = "Attrition Distribution by Gender", x = "Gender", y = "Count") +

scale\_fill\_manual(values = c("lightblue", "pink")) +

theme\_minimal()



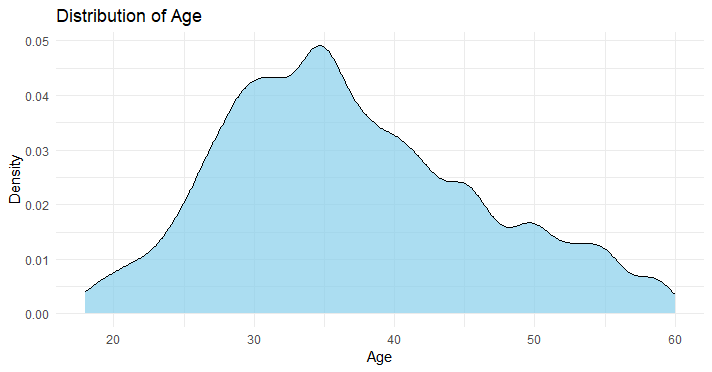
**Distribution of Age**

ggplot(data, aes(x = Age)) +

geom\_density(fill = "skyblue", alpha = 0.7) +

labs(title = "Distribution of Age", x = "Age", y = "Density") +

theme\_minimal()



**Multiple Distribution Plots**

vars<-c("TotalWorkingYears","MonthlyIncome","YearsAtCompany", "DistanceFromHome","YearsInCurrentRole","YearsWithCurrManager", "YearsSinceLastPromotion","PercentSalaryHike","TrainingTimesLastYea")

plots <- lapply(vars, function(var) {

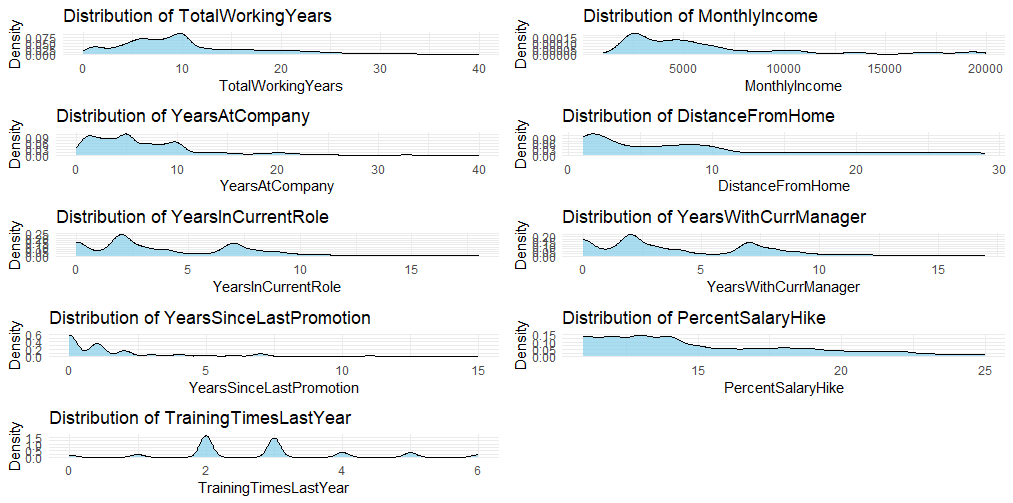
ggplot(data, aes\_string(x = var)) +

geom\_density(fill = "skyblue", alpha = 0.7) +

labs(title = paste("Distribution of", var), x = var, y = "Density") + theme\_minimal()

})

do.call(grid.arrange, c(plots, ncol = 2))



**Different categorical distributions of data:**

categorical <- c('Attrition', 'BusinessTravel', 'Department', 'EducationField', 'Gender', 'OverTime', 'Employee.Source', 'MaritalStatus')

categorical\_plots <- lapply(categorical[-1], function(cat\_var) {

ggplot(data, aes\_string(x = cat\_var, fill = "Attrition")) +

geom\_bar(position = "dodge") +

labs(title = paste("Attrition by", cat\_var), x = cat\_var, y = "Count") +

scale\_fill\_manual(values = c("lightblue", "pink")) +

theme\_minimal()

})

do.call(grid.arrange, c(categorical\_plots, ncol = 2))



**Split Dat into Training and Test Sets**

Splits the data into 80% training and 20% testing.

set.seed(123)

train\_index <- createDataPartition(data$Attrition, p = 0.8, list = FALSE)

train <- data[train\_index, ]

test <- data[-train\_index, ]

**RFE for feature selection:**

ctrl <- rfeControl(functions=rfFuncs, method="cv", number=10)

rfe\_results <- rfe(X, y, sizes=1:ncol(X), rfeControl=ctrl)

mean\_scores <- rfe\_results$results$Accuracy

n\_features <- rfe\_results$results$Variables

ggplot(data.frame(Features = n\_features, Accuracy = mean\_scores), aes(x = Features, y = Accuracy)) +

geom\_line(color = "blue", size = 1) +

geom\_point(color = "blue", size = 2) +

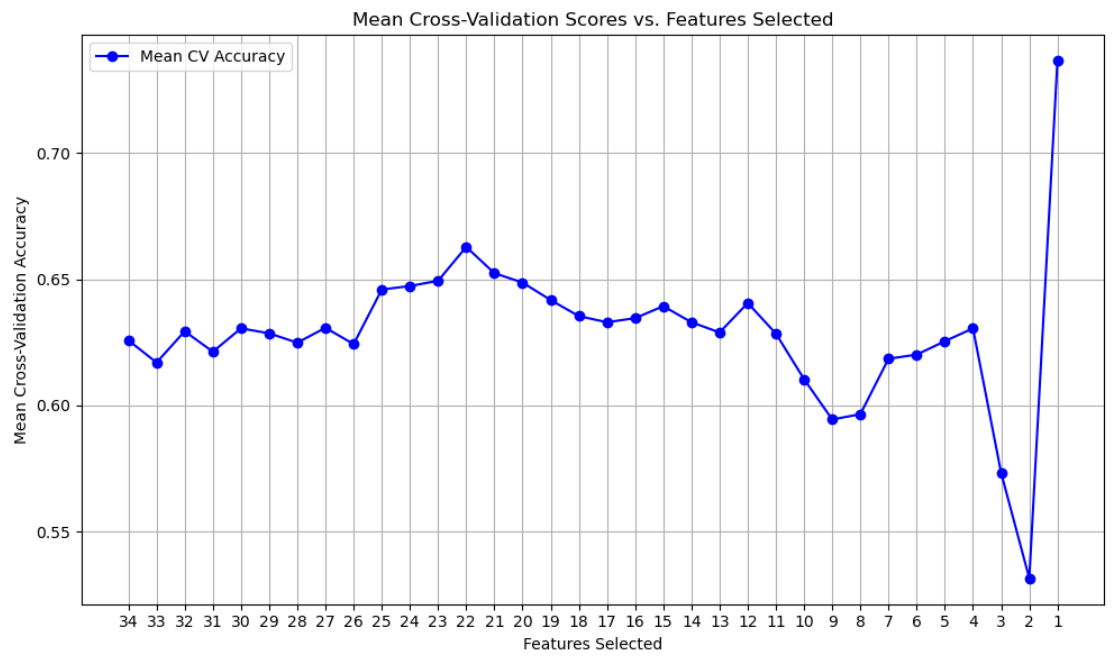
labs(title = "Mean Cross-Validation Scores vs. Features Selected",

x = "Number of Features Selected", y = "Mean CV Accuracy") +

theme\_minimal() +

scale\_x\_continuous(breaks = seq(1, max(n\_features), by = 1)) +

theme(axis.text.x = element\_text(angle = 90, hjust = 1))



**Feature Selection**

Selects the top 21 features based on domain knowledge or feature importance.

top\_features <- names(train)[1:21] # Replace with actual feature selection method

train\_top <- train %>% select(all\_of(top\_features), Attrition)

test\_top <- test %>% select(all\_of(top\_features), Attrition)

**Modeling and Evaluation**

**-Random Forest**

set.seed(123)

rf\_model <- randomForest(Attrition ~ ., data = train\_top, importance = TRUE)

rf\_predictions <- predict(rf\_model, newdata = test\_top)

rf\_accuracy <- confusionMatrix(rf\_predictions, test\_top$Attrition)$overall['Accuracy']

**-SVM**

set.seed(123)

svm\_model <- svm(Attrition ~ ., data = train\_top)

svm\_predictions <- predict(svm\_model, newdata = test\_top)

svm\_accuracy <- confusionMatrix(svm\_predictions, test\_top$Attrition)$overall['Accuracy']

**Model Comparison**

accuracy\_df <- data.frame(

Model = c("Random Forest", "SVM"),

Accuracy = c(rf\_accuracy, svm\_accuracy)

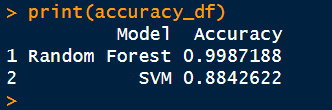
)

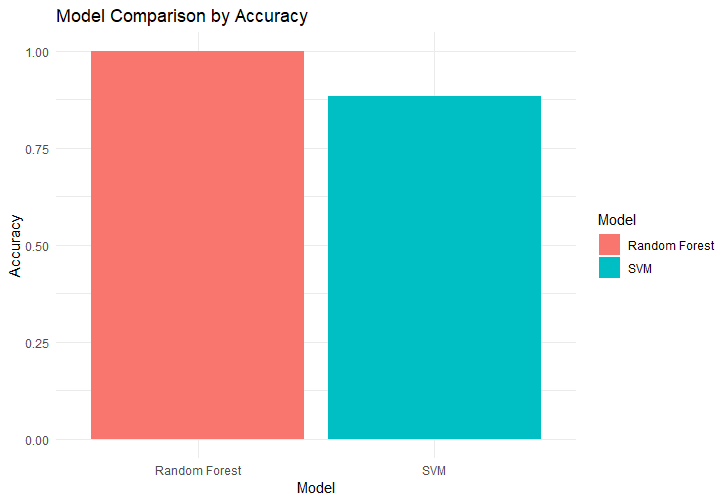
ggplot(accuracy\_df, aes(x = Model, y = Accuracy, fill = Model)) +

geom\_bar(stat = "identity") +

labs(title = "Model Comparison by Accuracy", x = "Model", y = "Accuracy") +

theme\_minimal()





**Best Model**

best\_model <- accuracy\_df[which.max(accuracy\_df$Accuracy), ]

print(paste("Best Model: ", best\_model$Model))



**Confusion Matrix for best model**

best\_predictions <- switch(

as.character(best\_model$Model),

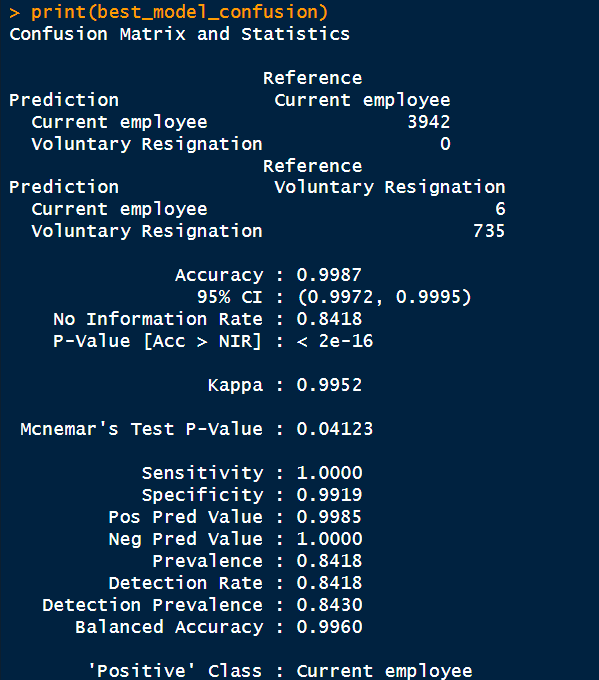
"Random Forest" = rf\_predictions,

"SVM" = svm\_predictions

)

best\_model\_confusion <- confusionMatrix(best\_predictions, test\_top$Attrition)

print(best\_model\_confusion)



**ROC Curves**

roc\_rf <- roc(as.numeric(test\_top$Attrition), as.numeric(rf\_predictions))

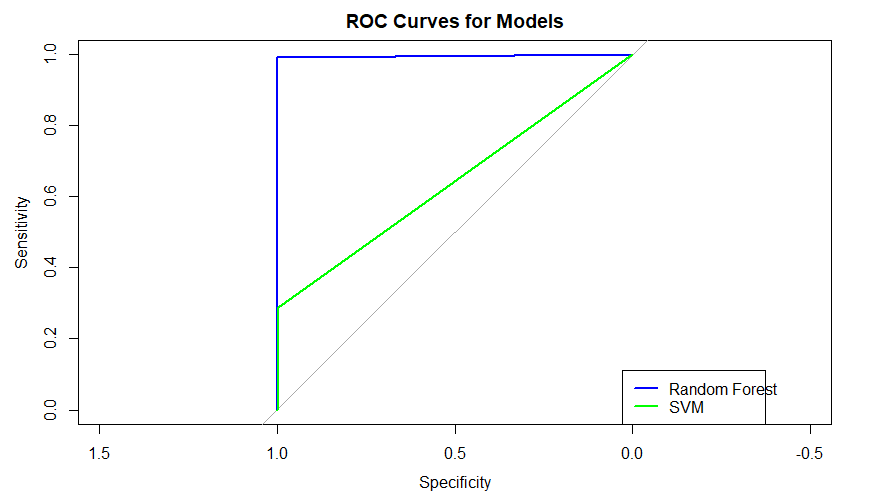
roc\_svm <- roc(as.numeric(test\_top$Attrition), as.numeric(svm\_predictions))

plot(roc\_rf, col = "blue", lwd = 2, main = "ROC Curves for Models")

lines(roc\_svm, col = "green", lwd = 2)

legend("bottomright", legend = c("Random Forest", "SVM"),

col = c("blue", "green"), lwd = 2)



### Concluding Remarks

**References**

Dataset is available at:

https://www.kaggle.com/datasets/sheenabatra/facebook-data Hive Programming reference is available at:

https://www.tutorialspoint.com/hive/index.htm Oracle VM VirtualBox is available at:

https://www.virtualbox.org/wiki/Downloads

Hortonworks Sandbox is available at:

https://www.cloudera.com/downloads/hdp.html Others:

https://unmetric.com/resources/facebook-data-analytics